Clustering in Melanoma Cancer Detection

Alana Jocelyn Natania Massie
Department of **Computer Science and Electronics**
Universitas Gadjah Mada
Yogyakarta 55281, Indonesia
alanajocelynnataniamassie@mail.ugm.ac.id

Vian Sebastian Bromokusumo
Department of **Computer Science and Electronics**
Universitas Gadjah Mada
Yogyakarta 55281, Indonesia
viansebastianbromokusumo@mail.ugm.ac.id

Argya Nur Adhirajasa
Department of **Computer Science and Electronics**
Universitas Gadjah Mada
Yogyakarta 55281, Indonesia
argyanuradhirajasa@mail.ugm.ac.id

Khalisha Fadiya Khansa
Department of **Computer Science and Electronics**
Universitas Gadjah Mada
Yogyakarta 55281, Indonesia
khalishafadiyakhansa@mail.ugm.ac.id

***Abstract*—The latest skin cancer data shows that Melanoma of skin is the 17th most common cancer worldwide. This report explores the application of KMeans clustering and MobileNetV2 for the classification of skin cancer images, addressing the need for effective diagnostic tools in dermatology. By implementing KMeans clustering, the project aims to segment skin lesion images into groups indicative of benign or malignant characteristics, enhancing the accuracy of preliminary diagnoses. The integration of the MobileNetV2 is used for its efficiency and high performance in extracting features. Additionally, Principal Component Analysis (PCA) is used to condense the feature set into a two-dimensional format for easier visualization and analysis. Results showed that clustering, particularly KMeans clustering, proved to be effective given the higher overall score for the segmented image cluster and the improved performance in CNN.**

Keywords—KMeans clustering, feature extraction, machine learning applications, skin cancer classification

I. INTRODUCTION

Skin cancer is a major public health concern, with millions of new cases occurring worldwide annually. According to the International Agency for Research on Cancer of the World Health Organization, 1.5 million new cases were identified in 2022, making it the most common group of cancer. It is estimated that in the same year, around 330,000 individuals received a melanoma diagnosis, with the disease resulting in almost 60,000 deaths globally. [1] This emphasizes the importance of accurate and early detection methods.

In recent years, machine learning algorithms have been utilized in diagnosing skin diseases [2]. Among various machine learning techniques, unsupervised learning algorithms such as KMeans clustering have shown potential in segmenting medical images. This project capitalizes on this potential by applying KMeans clustering to segment skin lesion images into groups indicative of benign or malignant characteristics.

The use of KMeans clustering in medical image analysis presents a promising approach to its efficiency and accuracy. However, its application in skin cancer detection comes with challenges. The variation of skin lesions in terms of color, size, and shape makes the task of segmentation and feature extraction harder, which requires a feature extraction process that can capture the nuances of skin lesion images.

This project introduces an approach to the classification of skin cancer images, using the abilities of KMeans clustering for preliminary image segmentation and the MobileNetV2 model for deep learning feature extraction. The selection of MobileNetV2 is motivated by its efficient yet powerful network architecture. [3]

Furthermore, Principal Component Analysis concerts a large data set into a more compact dimensional space, extracting the most informative features while maintaining the essential information from the original dataset [4]. The implementation of Principal Component Analysis for dimensionality reduction compresses the high dimensional feature space into a two-dimensional plane. This simplification enhances the visualization and understanding of the data clusters.

II. METHODS AND IMPLEMENTATION

For the implementation, there are nine stages, starting with the data acquisition and preprocessing, until the model training and evaluation, with various methods being used.

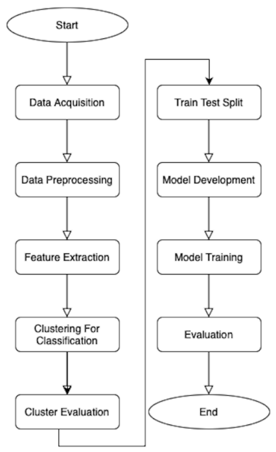

Fig. 1: Flowchart of KMeans Image Classification

A. Data Acquisition

The dataset used in this final project is from Kaggle [5] . This dataset contains roughly 13,900 images of benign and malignant melanoma cancer. For this final project, only 100 images from each class are used. The reason for this is to save time and memory, as the GPU service offered in Google Colaboratory is limited.

B. Preprocessing

Data preprocessing lets the users decide on how to represent the data, which concepts to learn, and how to present the results of the analyzed data so that it's easier to interpret. In practice, the first operation on any set of data is preprocessing. [6]

The preprocessing method used in this final project is image segmentation by KMeans Clustering. The k-means algorithm is the oldest and popular partitional method. K-means clustering is also known for its simplicity, making it the most commonly used clustering algorithm. [7]

The KMeans algorithm we constructed, used in all clustering applications in this project are as follows.

```
def self_kmeans(X, n_clusters, max_iter=100, tol=1e-4,
random_state=42):
    # Initialize centroids randomly
    np.random.seed(random_state)
    centroids = X[np.random.choice(X.shape[0], n_clusters,
replace=False)]

    for _ in range(max_iter):
        # Assign data points to the nearest centroid
        distances = np.sqrt(((X[:, np.newaxis] - centroids) **
2).sum(axis=-1))
        labels = np.argmin(distances, axis=1)

        # Update the centroids
        new_centroids = np.array([X[labels == i].mean(axis=0)
for i in range(n_clusters)])

        # Check for convergence
        if np.max(np.abs(new_centroids - centroids)) < tol:
            break

        centroids = new_centroids

    return labels, centroids
```

Firstly, the number of clusters must be determined. This is achieved by using the Elbow method as well as Silhouette Analysis. The visualization of them can be seen as follows.

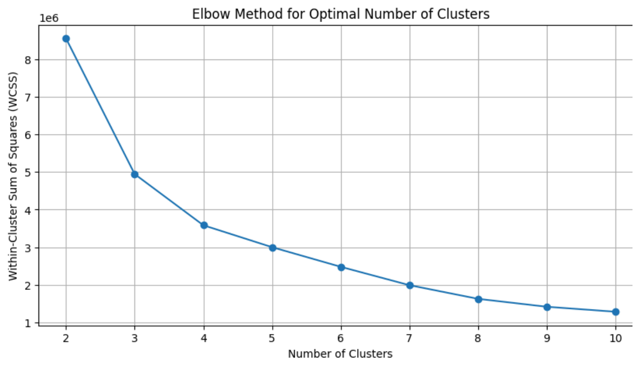

Fig. 2.1 : Elbow method output

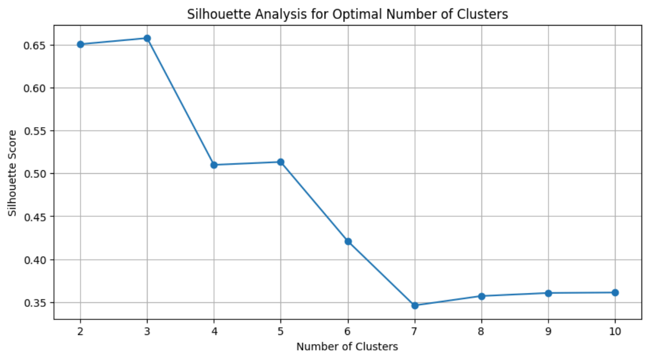

Fig. 2.2 : Silhouette Analysis output

Given the visualization above, we can infer that the optimal number of clusters based on the Elbow method is either 4 or 5, while the optimal number of clusters based on the Silhouette Analysis is 3. After a few tests, the number of clusters used will be 5, as it performs the segmentation but still preserves important features. The result of the clusters can be seen as follows.

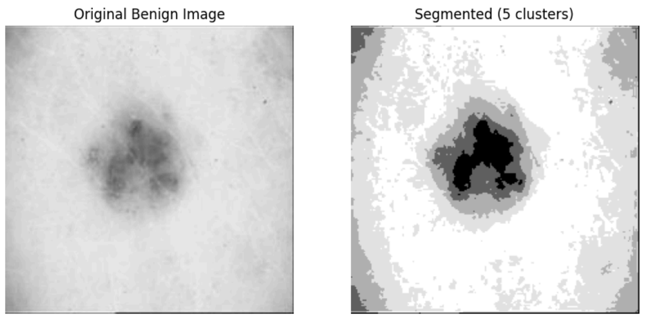

Fig. 2.3: Segmentation of Benign images

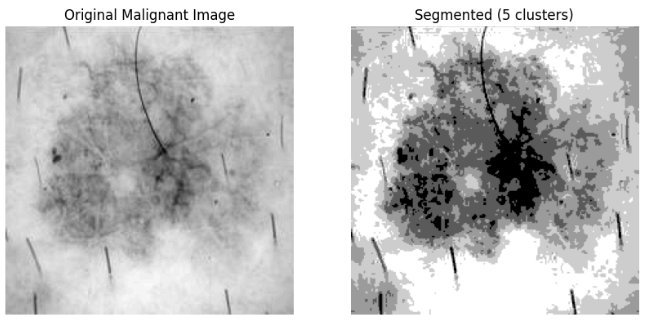

Fig. 2.4: Segmentation of Malignant images

C. Feature Extraction

In feature extraction, the features can be divided into four types; Geometric features, Statistical features, Texture features, and Color features. [8] For feature extraction in this project, a transfer learning approach is used using the MobileNetV2[9] pretrained model with imagenet as its weights.

The MobileNetV2 model was chosen due to its computational efficiency and high performance. The approach of using transfer learning allows leveraging the knowledge gained from training on a large dataset, in this case, imagenet, to extract meaningful features from the melanoma dataset.

The image weights (features) are then stored in a list as vector representations, and this list is then fed to the clustering algorithm, where similar features will be grouped together, classifying them. The source code of the function can be seen as follows.

```
# load pretrained model
base_model = MobileNetV2(weights = 'imagenet',
include_top = False, input_shape = (224, 224, 3))

for layer in base_model.layers:
    layer.trainable = False

def extract_vector(data_holder):
    mob_feature_list = []
    for im in data_holder:
        im = image.img_to_array(im)
        img = preprocess_input(np.expand_dims(im.copy(),
axis=0))
        mob_feature = base_model.predict(img)
        mob_feature_np = np.array(mob_feature)
        mob_feature_list.append(mob_feature_np.flatten())
    return np.array(mob_feature_list)
```

The output of the feature extraction can be seen as follows.

```
benign features represented in numpy array:
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]

malignant features represented in numpy array:
[[0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 ...
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]
 [0. 0. 0. ... 0. 0. 0.]]
```

D. Clustering for Classification

Clustering organizes data instances into distinct subsets based on their similarity, ensuring that similar instances are grouped together and dissimilar ones are allocated to different groups. This will result in an effective representation of the sampled population and lead to efficient organization and analysis of the data. [10]

To cluster the features into groups, the PCA method is used to reduce dimensionality, in this case into 2. The result of the cluster of each class are as follows.

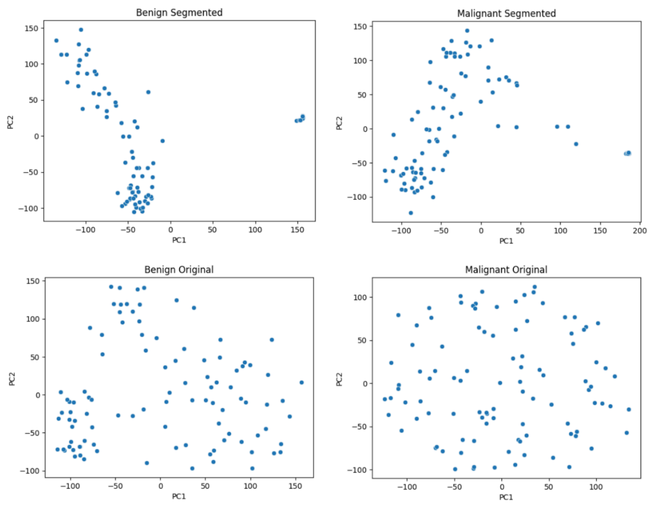

Fig. 3.0 : Representations of each class

Clusters of the segmented and original images can be seen as follows.

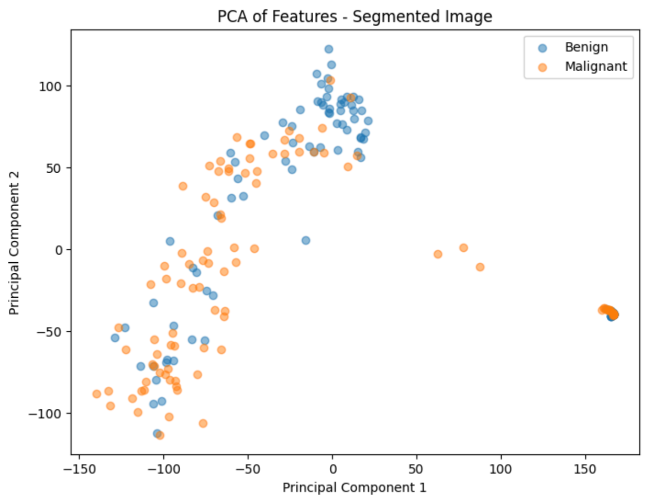

Fig. 3.1 : Clusters of segmented images

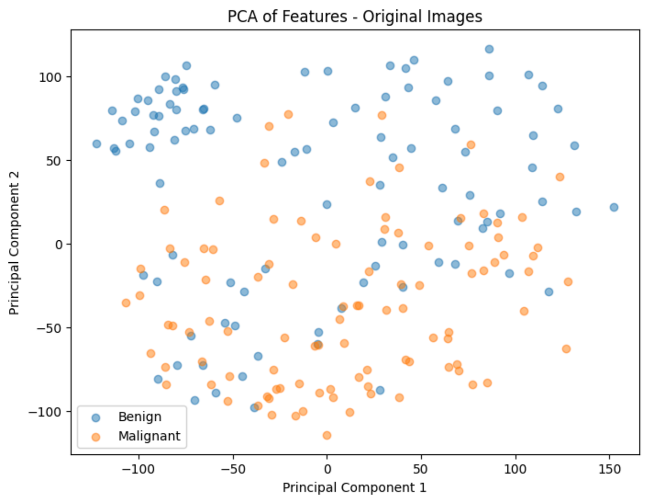

Fig. 3.2 : Clusters of original images

As can be seen in the visualizations, segmented images produce better formed clusters as a whole. The evaluation of these clusters will be discussed in a later section.

E. Validation with Convolutional Neural Network (CNN)

A widely recognized deep learning structure, the Convolutional Neural Network (CNN), draws inspiration from the innate visual perception mechanisms observed in living organisms. [11] Validation using CNN is implemented in this project to better observe how clustering aids in melanoma detection with steps as follows.

1. Train Test Split

   First, to ensure an unbiased model evaluation, the dataset, in this case both segmented and original images, is first split into training and testing sets with 80:20 ratio. The output of this split are:

   a. 160 train segmented images,
   b. 160 train original images,
   c. 40 test segmented images,
   d. 40 test original images.

   The dataset is now ready to be fed to the CNN model.

2. Model Development

   In developing the model, the same transfer learning model, the MobileNetv2, is used. In addition to that, a layer of Batch Normalization and Flatten, as well as two Dense output layers are added. This is done to perform a simple customization for the model to detect our dataset's features. The model's architecture can be seen as follows.

```
model = tf.keras.models.Sequential([
    base_model,
    BatchNormalization(),
    Flatten(),
    Dense(512, activation = 'relu'),
    Dense(1, activation='sigmoid')
])
```

   The model is then compiled with the Binary Cross Entropy loss function, as there are only two classes, the Adam optimizer function, with the default learning rate of 0.001, and the Accuracy metric. The model compilation code can be seen as follows.

```
model.compile(
    loss = 'binary_crossentropy',
    optimizer = tf.optimizers.Adam(),
    metrics = ['accuracy']
)
```

3. Model Training

   For the training process of the model, it is divided into two parts, training the dataset that has been processed with segmentation, as well as the original dataset. Both of the training processes were executed over 30 epochs.

III. Results and Discussion

I. Clustering

Evaluation for clustering will refer back to the result of image classifications between benign and malignant melanoma. The clustering is done twice, between images preprocessed with KMeans clustering and original images directly.

The evaluation methods used for clustering are purity and silhouette score. The silhouette coefficient is an internal evaluation method used to identify compact and well-separated clusters. It is a value between -1 and 1, where higher values indicate a better clustering. [12] Meanwhile, purity is an external evaluation measure. Purity scores indicate how consistent a cluster is in terms of its composition. When most of the labels within each cluster match, the purity score tends to be high. [13]

The functions for them are defined as follows.

```
def manual_silhouette_score(X, labels):
    n_samples = len(X)
    silhouette_values = np.zeros(n_samples)

    # Compute pairwise distances between
samples
    distances = np.zeros((n_samples, n_samples))
    for i in range(n_samples):
        for j in range(n_samples):
            distances[i, j] = np.linalg.norm(X[i] - X[j])

    for i in range(n_samples):
        # Compute average distance to other
samples in the same cluster (a_i)
        cluster_label = labels[i]
        cluster_indices = np.where(labels ==
cluster_label)[0]
        a_i = np.mean(distances[i, cluster_indices])

        # Compute average distance to samples in
other clusters (b_i)
        b_i = np.min([np.mean(distances[i,
np.where(labels == other_label)[0]]) for
other_label in np.unique(labels) if other_label !=
cluster_label])

        # Compute silhouette value for sample i
        silhouette_values[i] = (b_i - a_i) / max(a_i, b_i)

    # Compute mean silhouette score
    silhouette_score = np.mean(silhouette_values)
    return silhouette_score


def purity_score(y_true, y_pred):
    # Convert labels to integers
    y_true_int = np.asarray(y_true, dtype=int)
    y_pred_int = np.asarray(y_pred, dtype=int)

    # Compute contingency matrix
    contingency_matrix =
np.zeros((len(np.unique(y_true_int)),
len(np.unique(y_pred_int))))
    for i, j in zip(y_true_int, y_pred_int):
        contingency_matrix[i, j] += 1

    # Compute purity
    purity = np.sum(np.amax(contingency_matrix,
axis=0)) / np.sum(contingency_matrix)

    return purity
```

The results of the purity and silhouette score are as follows.

Cluster	Segmented Images Cluster	Original Images Cluster
Purity	0.695	0.715
Silhouette Score	0.187	0.054

Based on the results above, it can be observed that the segmented images cluster excels greatly in silhouette score, meaning the cluster and data can be differentiated very effectively. On the other hand, the silhouette score of the original images cluster is poor, and both of these can be observed in the visualization of fig. 3.1 and 3.2.

The results of the purity however, does not vary greatly, with only 0.02, where the original image cluster has the upper hand. This infers that the classification process is better. This may be due to the nature of image segmentation using clustering, as it groups similar features in the image, causing a slight loss in coherent features in the image.

Seeing the results, we can conclude that the overall score of the segmented image cluster is greater than the original image cluster.

II. *CNN*

The model's performance on both the segmented and original image was evaluated using a confusion matrix based on the prediction that was on the images, and also by computing the accuracy and the loss of the model. The result of the confusion matrix can be seen as follows.

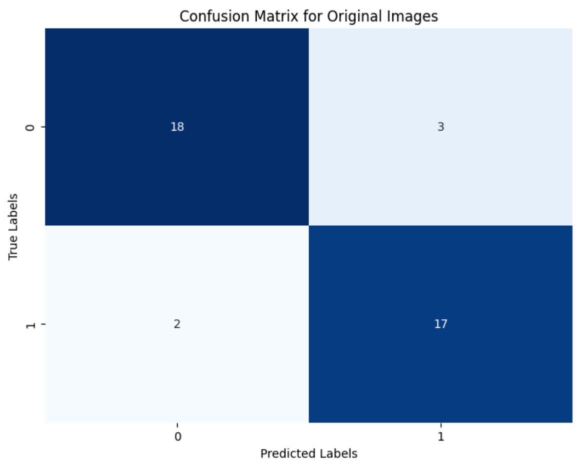

Fig. 4.1 : Confusion matrix for Original images

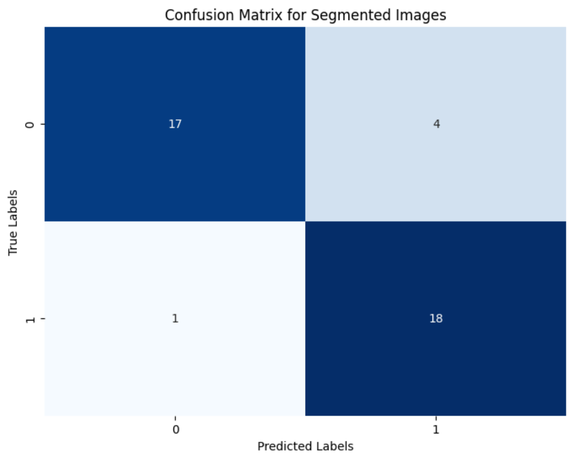

Fig. 4.2 : Confusion matrix for Segmented images

For the model with segmented images, the model achieved a loss of 1.39 and an accuracy of 0.875, while on the one with original images, it achieved a loss of 1.73 and an accuracy of 0.83. When comparing the two models, it can be seen that the model performed better on segmented images. The difference in performance proves that image segmentation, in this case, using KMeans clustering algorithm is capable of enhancing the model's ability to detect benign and malignant melanoma cancer.

IV. Conclusion

With the higher overall score for the segmented image cluster, as well as increase in performance in CNN, it can be observed the usefulness of clustering, especially KMeans clustering in aiding melanoma classification and detection. Although, it cannot be neglected that the accuracy and metrics reached are nowhere near the professional level. In future experimentation, a lot of improvements such as by using a larger dataset, incorporating physicians' diagnosis, implementing a higher level of preprocessing, model architecturing, and hyperparameter tuning, is highly recommended.

References

[1] International Agency for Research on Cancer, World Health Organization, "Skin Cancer," Available: https://www.iarc.who.int/cancer-type/skin-cancer/. Accessed on: Apr. 18, 2024.

[2] S. Salim and J. Al-Tuwaijari, "Skin Disease Classification System Based on Machine Learning Technique: A Survey," in IOP Conference Series: Materials Science and Engineering, vol. 1076, no. 1, Art. no. 012045, Feb. 2021, doi: 10.1088/1757-899X/1076/1/012045.

[3] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," presented at the Sixth Information Systems International Conference (ISICO 2021).

[4] E. Kavlakoglu, "Reducing Dimensionality with Principal Component Analysis with Python," IBM Developer, Mar. 06, 2024. [Online]. Available: https://developer.ibm.com/tutorials/awb-reducing-dimensionality-with-principal-component-analysis/. Accessed on: Apr. 19, 2024.

[5] Title: Bhavesh Mittal, "Melanoma Cancer Dataset," Kaggle, [2024]. [Online]. Available: https://www.kaggle.com/datasets/bhaveshmittal/melanoma-cancer-dataset

[6] A. Famili, W.-M. Shen, R. Weber, and E. Simoudis, "Data Preprocessing and Intelligent Data Analysis," Intelligent Data Analysis, vol. 1, no. 1, pp. 3–23, 1997. [Online]. Available: https://doi.org/10.3233/ida-1997-1102

[7] A. K. Jain and R. C. Dubes, "Algorithms for Clustering Data," Englewood Cliffs, NJ, USA: Prentice-Hall, 1988.

[8] Mutlag, W. K., Ali, S. K., Aydam, Z. M., & Taher, B. H. (2020). Feature Extraction Methods: A Review. Journal of Physics: Conference Series, 1591, 012028. doi:10.1088/1742-6596/1591/1/012028

[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 4510-4520, doi: 10.1109/CVPR.2018.00474.

[10] L. Rokach and O. Maimon, "Clustering Methods," in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Boston, MA: Springer, 2005, ch. 15. [Online]. Available: https://doi.org/10.1007/0-387-25465-X_15

[11] J. Gu et al., "Recent advances in convolutional neural networks," Pattern Recognition, vol. 77, pp. 354–377, 2018. doi:10.1016/j.patcog.2017.10.013

[12] P. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," J. Comput. Appl. Math., vol. 20, pp. 53-65, 1987.

[13] M. J. Bonder, S. Abeln, E. Zaura, and B. W. Brandt, "Comparing clustering and pre-processing in taxonomy analysis," Bioinformatics, vol. 28, no. 22, pp. 2891–2897, Nov. 2012. doi:10.1093/bioinformatics/bts552

Google Collaboratory Notebook link:
https://colab.research.google.com/drive/1oKpfETSJ2z6BGwKWpoqo46EXe7648Fge?usp=sharing

YouTube link:
https://youtu.be/SmwCyTNXYas?si=R3JsCPXzwADMldJn